WEST

Help | Logout | Interrupt

Main Menu | Search Form | Posting Counts | Show S Numbers | Edit S Numbers | Preferences | Cases

Search Results -

Terms	Documents
L17 and private	0

Database:
US Patents Full-Text Database
US Pre-Grant Publication Full-Text Database
JPO Abstracts Database
EPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Search:

Refine Search

Recall Text | Clear

Search History

DATE: Sunday, October 19, 2003 Printable Copy Create Case

Next page →

09/489,793

Set Name side by side	Query	Hit Count	Set Name result set
DB=USPT,JPAB,EPAB,DWPI,TDBD; PLUR=YES; OP=ADJ			
L18	L17 and private	0	L18
L17	L12 and reserved word$1	5	L17
L16	L12 and (public same protected)	0	L16
L15	L12 and (public same private)	9	L15
L14	L12 and (block$1 same public)	1	L14
L13	L12 and (block$1 same class$2)	10	L13
L12	L11 and structure edit$3	146	L12
L11	program$4 same structure same edit$3	1794	L11
L10	L8 and (private same order$3)	53	L10
L9	L8 and (private same order$3)	53	L9
L8	compil$3 same edit$3	2856	L8
L7	(pretty same edit$32)	32	L7
L6	WYSIWYG and (pretty same edit$32)	0	L6
L5	L3 and (public same private same protected same order$3)	20	L5
L4	L3 and (public same private same order$3)	125	L4
L3	program$4 same edit$3	22139	L3
L2	program$4 same edit$3 same preference order	2	L2
L1	program$4 same edit$3 same WYSIWYG	71	L1

END OF SEARCH HISTORY

Generate Collection | Print

Search Results - Record(s) 1 through 29 of 29 returned.

☐ 1. Document ID: US 6601057 B1

L20: Entry 1 of 29 File: USPT Jul 29, 2003

US-PAT-NO: 6601057
DOCUMENT-IDENTIFIER: US 6601057 B1

TITLE: Method and apparatus for generating and modifying multiple instances of an element of a web site

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw Desc | Image

☐ 2. Document ID: US 6549935 B1

L20: Entry 2 of 29 File: USPT Apr 15, 2003

US-PAT-NO: 6549935
DOCUMENT-IDENTIFIER: US 6549935 B1

TITLE: Method of distributing documents having common components to a plurality of destinations

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw Desc | Image

☐ 3. Document ID: US 6490604 B1

L20: Entry 3 of 29 File: USPT Dec 3, 2002

US-PAT-NO: 6490604
DOCUMENT-IDENTIFIER: US 6490604 B1

TITLE: Character information processing device equipped with a layout display function

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw Desc | Image

☐ 4. Document ID: US 6411299 B1

L20: Entry 4 of 29 File: USPT Jun 25, 2002

US-PAT-NO: 6411299
DOCUMENT-IDENTIFIER: US 6411299 B1

TITLE: Processing text for display on medical images

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw Desc | Image

☐ 5. Document ID: US 6256650 B1

L20: Entry 5 of 29 File: USPT Jul 3, 2001

US-PAT-NO: 6256650
DOCUMENT-IDENTIFIER: US 6256650 B1

TITLE: Method and system for automatically causing editable text to substantially occupy a text frame

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | KWIC | Draw Desc | Image

---

☐ 6. Document ID: US 6223191 B1

L20: Entry 6 of 29 File: USPT Apr 24, 2001

US-PAT-NO: 6223191
DOCUMENT-IDENTIFIER: US 6223191 B1
**** See image for Certificate of Correction ****

TITLE: Method and apparatus for automatically formatting multiple lines of text in a word processor

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | KWIC | Draw Desc | Image

---

☐ 7. Document ID: US 6202073 B1

L20: Entry 7 of 29 File: USPT Mar 13, 2001

US-PAT-NO: 6202073
DOCUMENT-IDENTIFIER: US 6202073 B1
**** See image for Certificate of Correction ****

TITLE: Document editing system and method

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | KWIC | Draw Desc | Image

---

☐ 8. Document ID: US 6098071 A

L20: Entry 8 of 29 File: USPT Aug 1, 2000

US-PAT-NO: 6098071
DOCUMENT-IDENTIFIER: US 6098071 A
**** See image for Certificate of Correction ****

TITLE: Method and apparatus for structured document difference string extraction

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | KWIC | Draw Desc | Image

---

☐ 9. Document ID: US 6014663 A

L20: Entry 9 of 29 File: USPT Jan 11, 2000

US-PAT-NO: 6014663
DOCUMENT-IDENTIFIER: US 6014663 A

** See image for Certificate Correction **

TITLE: System, method, and computer program product for comparing text portions by reference to index information

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | KWIC | Draw Desc | Image

---

☐ 10. Document ID: US 6014137 A

L20: Entry 10 of 29 File: USPT Jan 11, 2000

US-PAT-NO: 6014137
DOCUMENT-IDENTIFIER: US 6014137 A

TITLE: Electronic kiosk authoring system

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | KWIC | Draw Desc | Image

---

☐ 11. Document ID: US 6009196 A

L20: Entry 11 of 29 File: USPT Dec 28, 1999

US-PAT-NO: 6009196
DOCUMENT-IDENTIFIER: US 6009196 A

TITLE: Method for classifying non-running text in an image

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | KWIC | Draw Desc | Image

---

☐ 12. Document ID: US 5983227 A

L20: Entry 12 of 29 File: USPT Nov 9, 1999

US-PAT-NO: 5983227
DOCUMENT-IDENTIFIER: US 5983227 A

TITLE: Dynamic page generator

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | KWIC | Draw Desc | Image

---

☐ 13. Document ID: US 5963956 A

L20: Entry 13 of 29 File: USPT Oct 5, 1999

US-PAT-NO: 5963956
DOCUMENT-IDENTIFIER: US 5963956 A

TITLE: System and method of optimizing database queries in two or more dimensions

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | KWIC | Draw Desc | Image

---

☐ 14. Document ID: US 5956726 A

L20: Entry 14 of 29 File: USPT Sep 21, 1999

US-PAT-NO: 5956726
DOCUMENT-IDENTIFIER: US 5956726 A

TITLE: Method and apparatus for structured document difference string extraction

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | KWIC | Draw Desc | Image

---

☐ 15. Document ID: US 5953733 A

L20: Entry 15 of 29 File: USPT Sep 14, 1999

US-PAT-NO: 5953733
DOCUMENT-IDENTIFIER: US 5953733 A

TITLE: Electronic publishing system

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | KWIC | Draw Desc | Image

---

☐ 16. Document ID: US 5926185 A

L20: Entry 16 of 29 File: USPT Jul 20, 1999

US-PAT-NO: 5926185
DOCUMENT-IDENTIFIER: US 5926185 A

TITLE: Method for processing a set of page description language commands to reduce complexity

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | KWIC | Draw Desc | Image

---

☐ 17. Document ID: US 5920879 A

L20: Entry 17 of 29 File: USPT Jul 6, 1999

US-PAT-NO: 5920879
DOCUMENT-IDENTIFIER: US 5920879 A

TITLE: Document structure conversion apparatus

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | KWIC | Draw Desc | Image

---

☐ 18. Document ID: US 5907837 A

L20: Entry 18 of 29 File: USPT May 25, 1999

US-PAT-NO: 5907837
DOCUMENT-IDENTIFIER: US 5907837 A
**** See image for Certificate of Correction ****

TITLE: Information retrieval system in an on-line network including separate content and layout of published titles

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | KWIC | Draw Desc | Image

☐ 19. Document ID: US 5875265 A

L20: Entry 19 of 29 File: USPT Feb 23, 1999

US-PAT-NO: 5875265
DOCUMENT-IDENTIFIER: US 5875265 A

TITLE: Image analyzing and editing apparatus using psychological image effects

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | KWIC | Draw Desc | Image

☐ 20. Document ID: US 5841900 A

L20: Entry 20 of 29 File: USPT Nov 24, 1998

US-PAT-NO: 5841900
DOCUMENT-IDENTIFIER: US 5841900 A

TITLE: Method for graph-based table recognition

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | KWIC | Draw Desc | Image

☐ 21. Document ID: US 5802381 A

L20: Entry 21 of 29 File: USPT Sep 1, 1998

US-PAT-NO: 5802381
DOCUMENT-IDENTIFIER: US 5802381 A

TITLE: Text editor for converting text format to correspond to an output method

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | KWIC | Draw Desc | Image

☐ 22. Document ID: US 5708806 A

L20: Entry 22 of 29 File: USPT Jan 13, 1998

US-PAT-NO: 5708806
DOCUMENT-IDENTIFIER: US 5708806 A

TITLE: Data processing system and method for generating a representation for and for representing electronically published structured documents

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | KWIC | Draw Desc | Image

☐ 23. Document ID: US 5669007 A

L20: Entry 23 of 29 File: USPT Sep 16, 1997

US-PAT-NO: 5669007
DOCUMENT-IDENTIFIER: US 5669007 A

TITLE: Method and system for analyzing the logical structure of a document

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | KWIC | Draw Desc | Image

☐ 24. Document ID: US 5634016 A

L20: Entry 24 of 29 File: USPT May 27, 1997

US-PAT-NO: 5634016
DOCUMENT-IDENTIFIER: US 5634016 A
**** See image for Certificate of Correction ****

TITLE: Event management system

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | KWIC | Draw Desc | Image

☐ 25. Document ID: US 5600771 A

L20: Entry 25 of 29 File: USPT Feb 4, 1997

US-PAT-NO: 5600771
DOCUMENT-IDENTIFIER: US 5600771 A

TITLE: Method of document layout process and system therefor

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | KWIC | Draw Desc | Image

☐ 26. Document ID: US 5566289 A

L20: Entry 26 of 29 File: USPT Oct 15, 1996

US-PAT-NO: 5566289
DOCUMENT-IDENTIFIER: US 5566289 A

TITLE: Document formatting support system

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | KWIC | Draw Desc | Image

☐ 27. Document ID: US 5555362 A

L20: Entry 27 of 29 File: USPT Sep 10, 1996

US-PAT-NO: 5555362
DOCUMENT-IDENTIFIER: US 5555362 A

TITLE: Method and apparatus for a layout of a document image

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | KWIC | Draw Desc | Image

☐ 28. Document ID: US 5438512 A

L20: Entry 28 of 29 File: USPT Aug 1, 1995

US-PAT-NO: 5438512

DOCUMENT-IDENTIFIER: US 54385 A

TITLE: Method and apparatus for specifying layout processing of structured documents

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | KWIC | Draw Desc | Image

---

☐ 29. Document ID: US 4813010 A

L20: Entry 29 of 29 File: USPT Mar 14, 1989

US-PAT-NO: 4813010
DOCUMENT-IDENTIFIER: US 4813010 A

TITLE: Document processing using heading rules storage and retrieval system for generating documents with hierarchical logical architectures

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | KWIC | Draw Desc | Image

---

Generate Collection | Print

Terms	Documents
L19 and (structur$2 same edit$3)	29

Display Format: - Change Format

Previous Page Next Page

Generate Collection | Print

Search Results - Record(s) 1 through 10 of 10 returned.

☐ 1. Document ID: US 6246404 B1

L13: Entry 1 of 10 File: USPT Jun 12, 2001

US-PAT-NO: 6246404
DOCUMENT-IDENTIFIER: US 6246404 B1
**** See image for Certificate of Correction ****

TITLE: Automatically generating code for integrating context-sensitive help functions into a computer software application

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments KWIC | Draw Desc | Image

☐ 2. Document ID: US 6141596 A

L13: Entry 2 of 10 File: USPT Oct 31, 2000

US-PAT-NO: 6141596
DOCUMENT-IDENTIFIER: US 6141596 A

TITLE: Fieldbus network configuration utility with improved scheduling and looping

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments KWIC | Draw Desc | Image

☐ 3. Document ID: US 5971581 A

L13: Entry 3 of 10 File: USPT Oct 26, 1999

US-PAT-NO: 5971581
DOCUMENT-IDENTIFIER: US 5971581 A

TITLE: Fieldbus network configuration utility with improved scheduling and looping

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments KWIC | Draw Desc | Image

☐ 4. Document ID: US 5862395 A

L13: Entry 4 of 10 File: USPT Jan 19, 1999

US-PAT-NO: 5862395
DOCUMENT-IDENTIFIER: US 5862395 A

TITLE: Customizable user interfaces for programmed computer systems

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments KWIC | Draw Desc | Image

☐ 5. Document ID: US 5857212 A

L13: Entry 5 of 10 File: USPT Jan 5, 1999

US-PAT-NO: 5857212
DOCUMENT-IDENTIFIER: US 5857212 A

TITLE: System and method for horizontal alignment of tokens in a structural representation program editor

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | KWIC | Draw Desc | Image

---

☐ 6. Document ID: US 5813019 A

L13: Entry 6 of 10 File: USPT Sep 22, 1998

US-PAT-NO: 5813019
DOCUMENT-IDENTIFIER: US 5813019 A
**** See image for Certificate of Correction ****

TITLE: Token-based computer program editor with program comment management

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | KWIC | Draw Desc | Image

---

☐ 7. Document ID: US 5752058 A

L13: Entry 7 of 10 File: USPT May 12, 1998

US-PAT-NO: 5752058
DOCUMENT-IDENTIFIER: US 5752058 A
**** See image for Certificate of Correction ****

TITLE: System and method for inter-token whitespace representation and textual editing behavior in a program editor

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | KWIC | Draw Desc | Image

---

☐ 8. Document ID: US 5748975 A

L13: Entry 8 of 10 File: USPT May 5, 1998

US-PAT-NO: 5748975
DOCUMENT-IDENTIFIER: US 5748975 A
**** See image for Certificate of Correction ****

TITLE: System and method for textual editing of structurally-represented computer programs with on-the-fly typographical display

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | KWIC | Draw Desc | Image

---

☐ 9. Document ID: US 5740425 A

L13: Entry 9 of 10 File: USPT Apr 14, 1998

US-PAT-NO: 5740425

DOCUMENT-IDENTIFIER: US 57404[redacted] A
**** See image for Certificate of Correction ****

TITLE: Data structure and method for publishing electronic and printed product catalogs

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | KWIC | Draw Desc | Image

☐ 10. Document ID: US 5737608 A

L13: Entry 10 of 10 File: USPT Apr 7, 1998

US-PAT-NO: 5737608
DOCUMENT-IDENTIFIER: US 5737608 A
**** See image for Certificate of Correction ****

TITLE: Per-keystroke incremental lexing using a conventional batch lexer

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | KWIC | Draw Desc | Image

Generate Collection | Print

Terms	Documents
L12 and (block$1 same class$2)	10

Display Format: - Change Format

Previous Page Next Page

Search Results for: [C++ AND structure editors]
Found **261** of **121,820** searched.

(10/19/2003)

Warning: Maximum result set of 200 exceeded. Consider refining.

Search within Results

> Advanced Search

> Search Help/Tips

Sort by: Title Publication Publication Date Score

Results 1 - 20 of 200 short listing

Prev Page 1 2 3 4 5 6 7 8 9 10 Next Page

1 Interactive Editing Systems: Part II 96%
Norman Meyrowitz , Andries van Dam
ACM Computing Surveys (CSUR) September 1982
Volume 14 Issue 3

2 Structured Editor Support for Modularity and Data Abstraction 95%
Michael Caplinger
ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 85 symposium on Language issues in programming environments June 1985
Volume 20 Issue 7
Language facilities for data abstraction and modularity have traditionally involved extensions to the declaratory syntax of the languages. Batch compiler technology forces the separation of the editing of this syntax from its being checked by the compiler, making programming more difficult. We discuss an alternate method of supporting such features in a language-based structured editor. Rather than adding syntax to the language, we simply use the editor to restrict uses of names. ...

3 Experience with an uncommon Lisp 95%
Cyril N. Alberga , Chris Bosman-Clark , Martin Mikelsons , Mary S. Van Deusen , Julian Padget
Proceedings of the 1986 ACM conference on LISP and functional programming August 1986

4 Preliminary experience from the dice system a distributed incremantal compiling environment 92%
Peter Fritzson
Proceedings of the first ACM SIGSOFT/SIGPLAN software engineering symposium on Practical software development environments April 1984

09/489,793

Volume 9 , 19 Issue 3 , 5
Abstracts The DICE system Is a highly Integrated programming environment which provides programmer support in the case where the programming environment resides in a host computer and the program Is running on a target computer that Is connected to the host. Such a system configuration is also suitable for remote debugging and maintenance of production versions of programs that has been installed in a user environment. The system contains tools such as an screen-oriented structur ...

5 Efficient abstractions for the implementation of structured editors 89%

Robert Hood

ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 85 symposium on Language issues in programming environments June 1985

Volume 20 Issue 7

This paper investigates the use of abstract recursive data structures and operations in the implementation of a structured program editor. The value-oriented semantics of the proposed constructs simplify the implementation of important features such as version control and an unbounded undo operation. Since the constructs can be implemented efficiently, their use in the structured program editor does not significantly affect its performance.

6 Dialogues: A basis for constructing programming environments 88%

John T. O'Donnell

ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 85 symposium on Language issues in programming environments June 1985

Volume 20 Issue 7

The components of a programming environment must communicate with the user while maintaining a state that is constantly evolving. We introduce the “dialogue”, an abstraction of such components, and we implement a dialogue function in a purely applicative language. The dialogue function exploits the properties of lasy evaluation and recursion to implement communication and state through recursively defined streams. We show how to define programming environment components using di ...

7 An advanced full-text retrieval and analysis system 88%

J. Smith , S. Weiss , G. Ferguson

Proceedings of the 10th annual international ACM SIGIR conference on Research and development in information retrieval November 1987

MICROARRAS is an advanced full-text retrieval and analysis system. It supports fast, efficient browsing of a document's vocabulary as well as its text, recursive analytic categories, Boolean search with flexible context specifications, evaluation of arithmetic expressions, and graphical display of various numeric distributions. The system is designed to work with large textbases stored on remote mainframes or on a local store for a micro-computer or workstation. The description covers syste ...

8 Generation of formatters for context-free languages 88%

Mark van den Brand , Eelco Visser

ACM Transactions on Software Engineering and Methodology (TOSEM) January 1996

Volume 5 Issue 1

Good documentation is important for the production of reusable and maintainable software. For the production of accurate documentation it is necessary that the original program text is not copied manually to obtain a typeset version. Apart from being tedious, this will invariably introduce errors. The production of tools that support the production of legible and accurate documentation is a software engineering challenge in itself. We present an algebraic approach to the generation of tools ...

9 Issues in the design of computer support for co-authoring and commenting 87%

Christine M. Neuwirth , David S. Kaufer , Ravinder Chandhok , James H. Morris

Proceedings of the 1990 ACM conference on Computer-supported cooperative work September 1990

This paper reports on a project to develop a “work in preparation” editor, or PREP editor, to study co-authoring and commenting relationships. As part of the project, we have identified three issues in designing computer support for co-authoring and commenting: (1) support for social interaction among co-authors and commenters; (2) support for cognitive aspects of co-authoring and external commenting; and (3) support for practicality in both types of interaction. For each of the ...

10 Programming on an already full brain 87%

Christopher Fry

Communications of the ACM April 1997

Volume 40 Issue 4

11 A small contribution to editing with a syntax directed editor 85%

Marvin V. Zelkowits

Proceedings of the first ACM SIGSOFT/SIGPLAN software engineering symposium on Practical software development environments April 1984

Volume 19 , 9 Issue 5 , 3

The simple design of a syntax directed editor has made them a popular research topic over the last few years; however, in practice little development time is spent in actually entering program text. Most of the effort is devoted to program maintenance. This means that the editor has to be able to handle changes easily - a task that most syntax editors handle poorly. This paper describes a syntax editor and outlines the features that make editing such programs practical. It includes a bottom ...

12 Reuse of compiler analysis in a programming environment 85%

M. P. Blivens , M. L. Soffa

Proceedings of the seventeenth annual ACM conference on Computer science : Computing trends in the 1990's: Computing trends in the 1990's February 1989

Productivity in the development of software can be increased by reusing code and design analysis. Following this approach we have developed an incremental optimizing compiler that reuses target code and compiler analysis. In order to be practical, it shares a database of information with other tools in a programming environment. The analysis performed by a compiler is reused to greatly reduce the recompilation time during program development and to incrementally produce target code that is ...

13 Centaur: the system 85%

P. Borras , D. Clement , Th. Despeyroux , J. Incerpi , G. Kahn , B. Lang , V. Pascual

ACM SIGPLAN Notices , Proceedings of the third ACM SIGSOFT/SIGPLAN software engineering symposium on Practical software development environments January 1989

Volume 24 Issue 2

This paper describes the organization of the CENTAUR system and its main components. The system is a generic interactive environment. When given the formal specification of a particular programming language-including syntax and semantics — it produces a language specific environment. This resulting environment includes a structure editor, an interpreter/debugger and other tools, all of which have graphic interfaces. CENTAUR is made of three parts: a database component, that provides s ...

14 An environment for developing adaptive, multi-device user interfaces 84%

John Grundy , Biao Yang

Proceedings of the Fourth Australian user interface conference on User interfaces 2003 - Volume

18 February 2003

There is a growing demand for the development of multi-device, adaptive user interfaces - interfaces that will run on and adapt to the characteristics of multiple display devices and networks as well as multiple users and user tasks. We describe a design and implementation environment for the development of such interfaces. This tool allows developers to specify their desired interfaces using an abstract set of screen element and layout constructs. It then generates a Java Server Page implementa ...

15 An Incremental Programming Environment 84%

Peter H. Feiler , Raul Medina-Mora

Proceedings of the 5th international conference on Software engineering March 1981

This document describes an Incremental Programming Environment (IPE) based on compilation technology, but providing facilities traditionally found only in interpretive systems. IPE provides a comfortable environment for a single programmer working on a single program. In IPE the programmer has a uniform view of the program in terms of the programming language. The program is manipulated through a syntax-directed editor and its execution is controlled by a debugging facility, which is integr ...

16 The role of modularity in document authoring systems 84%

Janet H. Walker

Proceedings of the ACM conference on Document processing systems January 2000

17 The impact of interprocedural analysis and optimization in the R^n programming environment 84%

Keith D. Cooper , Ken Kennedy , Linda Torczon

ACM Transactions on Programming Languages and Systems (TOPLAS) August 1986

Volume 8 Issue 4

In spite of substantial progress in the theory of interprocedural data flow analysis, few practical compiling systems can afford to apply it to produce more efficient object programs. To perform interprocedural analysis, a compiler needs not only the source code of the module being compiled, but also information about the side effects of every procedure in the program containing that module, even separately compiled procedures. In a conventional batch compiler system, the increase in compil ...

18 Corrigenda: “Pipeline Architecture” 84%

C. V. Ramamoorthy , H. F. Li

ACM Computing Surveys (CSUR) December 1978

Volume 10 Issue 4

19 Development concerns for a software design quality expert system 84%

Christopher W. Pidgeon , Peter A. Freeman

Proceedings of the 22nd ACM/IEEE conference on Design automation June 1985

This paper presents some developmental concerns for an expert system in the domain of software design. We discuss the context for such a system and explore the synergism between the human designer and automaton. A gross architecture for the automaton is given. A scenario of interaction between the designer and automaton is presented.

20 A dynamically self-adjusting structured editor 84%

Nazim H. Madhavji , Nikos Leoutsarakos

Proceedings of the 1985 ACM SIGSMALL symposium on Small systems May 1985

Two major characteristics shared by most program editors are that they are static, as they operate within a single framework of a full program text, and that they are strictly language based. These

characteristics are considered t[redacted]e less than desirable, as they appear to [redacted]rict the development of production software. As a solution, a new kind of structured editor, based on program fragments, is proposed in this paper. This kind of editor is dynamic, as ...

Search Results

Search Results for: [C++ AND pretty printers]
Found **159** of **121,820 searched.**

(10/18/2003)

Search within Results

GO > Advanced Search

> Search Help/Tips

Sort by: Title Publication Publication Date Score Binder

Results 1 - 20 of 159 short listing

Prev Page 1 2 3 4 5 6 7 8 Next Page

1 Automated deductive requirements analysis of critical systems 97%

ACM Transactions on Software Engineering and Methodology (TOSEM) July 2001
Volume 10 Issue 3
We advocate the need for automated support to System Requirement Analysis in the development of time- and safety-critical computer-based systems. To this end we pursue an approach based on deductive analysis: high-level, real-world entities and notions, such as events, states, finite variability, cause-effect relations, are modeled through the temporal logic TRIO, and the resulting deductive system is implemented by means of the theorem prover PVS. Throughout the paper, the constructs and f ...

2 Centaur: the system 91%

P. Borras , D. Clement , Th. Despeyroux , J. Incerpi , G. Kahn , B. Lang , V. Pascual
ACM SIGPLAN Notices , Proceedings of the third ACM SIGSOFT/SIGPLAN software engineering symposium on Practical software development environments January 1989
Volume 24 Issue 2
This paper describes the organization of the CENTAUR system and its main components. The system is a generic interactive environment. When given the formal specification of a particular programming language-including syntax and semantics — it produces a language specific environment. This resulting environment includes a structure editor, an interpreter/debugger and other tools, all of which have graphic interfaces. CENTAUR is made of three parts: a database component, that provides s ...

3 Document structure and modularity in mentor 90%

V. Donzeau-Gouge , G. Kahn , B. Lang , B. Mélèse
Proceedings of the first ACM SIGSOFT/SIGPLAN software engineering symposium on Practical software development environments April 1984
Volume 19 , 9 Issue 5 , 3
Mentor is a structured document manipulation system. It has been used for several years as a program

(09/489,793)

development and maintenance [redacted]ironment. Its main characteristics are: [redacted]s both interactive and programmable, it is parameterized by the language to be manipulated, it can manipulate several languages at the same time, as well as multi-lingual documents. it is open to and from the outer system, it is extensible. T ...

4 Conservative pretty printing 88%

Martin Ruckert

ACM SIGPLAN Notices February 1997

Volume 32 Issue 2

The main purpose of the pretty printing tool, presented here, is not so much the complete reformating of a program but is described better as finding and correcting formating errors.A set of relations is given describing the proper alignment of a piece of code in accordance with its syntactic structure. It is complemented by a simple algorithm to change any given source code to satisfy these relations.The relations are flexible enough to allow a variety of different formating styles while reject ...

5 Parsers and printers as stream destructors and constructors embedded in functional languages 87%

Michel Mauny

Proceedings of the fourth international conference on Functional programming languages and computer architecture November 1990

6 Tool support for refactoring functional programs 87%

Huiqing Li , Claus Reinke , Simon Thompson

Proceedings of the ACM SIGPLAN workshop on Haskell August 2003

Refactorings are source-to-source program transformations which change program structure and organisation, but not program functionality. Documented in catalogues and supported by tools, refactoring provides the means to adapt and improve the design of existing code, and has thus enabled the trend towards modern agile software development processes. Refactoring has taken a prominent place in software development and maintenance, but most of this recent success has taken place in the OO and XP co ...

7 Generation of formatters for context-free languages 87%

Mark van den Brand , Eelco Visser

ACM Transactions on Software Engineering and Methodology (TOSEM) January 1996

Volume 5 Issue 1

Good documentation is important for the production of reusable and maintainable software. For the production of accurate documentation it is necessary that the original program text is not copied manually to obtain a typeset version. Apart from being tedious, this will invariably introduce errors. The production of tools that support the production of legible and accurate documentation is a software engineering challenge in itself. We present an algebraic approach to the generation of tools ...

8 Growing languages with metamorphic syntax macros 85%

Claus Brabrand , Michael I. Schwartzbach

ACM SIGPLAN Notices , Proceedings of the 2002 ACM SIGPLAN workshop on Partial evaluation and semantics-based program manipulation January 2002

Volume 37 Issue 3

*"From now on, a main goal in designing a language should be to plan for growth." Guy Steele: Growing a Language, OOPSLA '98 invited talk.*We present our experiences with a syntax macro language which we claim forms a general abstraction mechanism for growing (domain-specific) extensions of programming languages. Our syntax macro language is designed to guarantee *type safety* and *termination*.A concept of *metamorphisms* allows the arguments of a macro to be inductively def ...

9 Z - the 95% program editor 84%

Steven R. Wood

Proceedings of the ACM SIGPLAN SIGOA symposium on Text manipulation June 1981

Recently much attention has been focused on structure-oriented program editors that have specific knowledge about the syntax and semantics of a particular programming language [1, 4, 5, 18]. These editors provide many desirable features for editing programs. However, the user interface is constrained by the syntax and semantics of the target language, and editing operations that are simple in a text editor can be quite complicated in a structure-oriented editor. In addition, the user has an ...

10 Multimedia and visualization: Approaches to comprehension-preserving graphical reduction of program visualizations 84%

F. Naharro-Berrocal , C. Pareja-Flores , J. Urquiza-Fuentes , J. Á. Velázquez-Iturbide

Proceedings of the 2002 ACM symposium on Applied computing March 2002

Past research efforts on the educational effectiveness of software animations agree in the necessity of active involvement of users, i.e. students and teachers. However, one of the main obstacles is the technical difficulty to produce them. Our approach seeks to generate software animations analogously to the generation of documents in office applications. The availability of static visualizations allows the user to define animations friendly; thus, he/she can select the most relevant ones to il ...

11 User Format Control in a LISP Prettyprinter 84%

Richard C. Waters

ACM Transactions on Programming Languages and Systems (TOPLAS) October 1983

Volume 5 Issue 4

12 Prettyprinting 84%

Dereck C. Oppen

ACM Transactions on Programming Languages and Systems (TOPLAS) October 1980

Volume 2 Issue 4

An algorithm for prettyprinting is given. For an input stream of length n and an output device with linewidth m, the algorithm requires time O(n) and space O(m). The algorithm is described in terms of two parallel processes: the first scans the input stream to determine the space required to print logical blocks of tokens; the second uses this information to decide where to break lines of t ...

13 An Ada interface to DIANA for inter-tool communication 84%

C. Howell , H. Gill , A. Harrison , D. Hough , T. Reed , T. Smith

Proceedings of the fifth Washington Ada symposium on Ada July 1988

14 A new approach to generic functional programming 84%

Ralf Hinze

Proceedings of the 27th ACM SIGPLAN-SIGACT symposium on Principles of programming languages January 2000

This paper describes a new approach to generic functional programming, which allows us to define functions generically for all datatypes expressible in Haskell. A generic function is one that is defined by induction on the structure of types. Typical examples include pretty printers, parsers, and comparison functions. The advanced type system of Haskell presents a real challenge: datatypes may be parameterized not only by types but also by type constructors, type definitions may involve mut ...

15 Implicit parameters: dynamic scoping with static types 84%

Jeffrey R. Lewis , John Launchbury , Erik Meijer , Mark B. Shields
Proceedings of the 27th ACM SIGPLAN-SIGACT symposium on Principles of programming languages January 2000
This paper introduces a language feature, called implicit parameters, that provides dynamically scoped variables within a statically-typed Hindley-Milner framework. Implicit parameters are lexically distinct from regular identifiers, and are bound by a special with construct whose scope is dynamic, rather than static as with let. Implicit parameters are treated by the type system as parameters that are not explicitly declared, but are inferred from their use. We ...

16 Moses: a graphics oriented software development environment 84%
G. Blaschek , G. Pomberger
Proceedings of the 15th annual conference on Computer Science February 1987
This paper describes a workstation-based monolingual programming environment which supports design, implementation, documentation and maintenance within the software production process. The most important features are graphic oriented program development, the design of the user interface and the adaptability to various programming languages.

17 The impact of computer-aided software engineering on student performance 84%
Mary J. Granger , Roger A. Pick
ACM SIGCSE Bulletin , Proceedings of the twenty-second SIGCSE technical symposium on Computer science education March 1991
Volume 23 Issue 1

18 A taxonomy for programming style 84%
Paul W. Oman , Curtis R. Cook
Proceedings of the 1990 ACM annual conference on Cooperation January 1990
Programming style guidelines, style analyzers, and code formatters have been developed without a solid empirical or theoretical basis. In this paper we provide: (1) a justification for developing a programming style taxonomy, (2) an operational style taxonomy, (3) example applications of the taxonomy illustrating the diverse and sometimes contradictory nature of programming style guidelines, and (4) a discussion on how the taxonomy can be used to further teaching and research in programming ...

19 A distributed architecture for programming environments 82%
Dominique Clément
ACM SIGSOFT Software Engineering Notes , Proceedings of the fourth ACM SIGSOFT symposium on Software development environments October 1990
Volume 15 Issue 6
Programming environments are typically based on concepts, such as syntax and semantics, and they provide functionalities, such as parsing, editing, type-checking, and compiling. Most existing programming environments are designed in a fully integrated manner, where parsers, editors, and semantic tools are tightly coupled. This leads to systems that are the sum of all their components, with obvious implications in terms of size, reusability, and maintainability. In this paper, we present a p ...

20 ISDE: An Interactive Software Development Environment 82%
M. Chesi , E. Dameri , M. P. Franceschi , M. G. Gatti , C. Simonelli
Proceedings of the first ACM SIGSOFT/SIGPLAN software engineering symposium on Practical software development environments April 1984
Volume 19 , 9 Issue 5 , 3
In this paper we present ISDE, an experimental Interactive Software Development Environment.

Briefly described is the design [redacted]thodology for the derivation of an inte[redacted]ive software development environment from a set of general meta-tools and (syntactic and semantic) language definition. The internal program representation common to every environment tool is presented, along with the main features of each implemented or under development tool, emphasizing tools integration and composition ...

Results 1 - 20 of 159 short listing

Prev Page 1 2 3 4 5 6 7 8 Next Page
